UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/670,026	09/24/2003	Christopher Philip Ruemmler	200311053-1	4526

22879 7590 06/01/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
KAWSAR, ABDULLAH AL	
ART UNIT	**PAPER NUMBER**
2109	

MAIL DATE	DELIVERY MODE
06/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/670,026	RUEMMLER ET AL.	
	Examiner	Art Unit	
	Abdullah-Al Kawsar	2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *9/24/2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *9/24/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 29 are pending.

Claim Objections

2. Claims 2, 9, 10, 12 and 13 are objected to because of the following informalities: claim 2 insert "and" in line 3 after priority register and claims 9, 10, 12 and 13 "IPF" and "PIC" should spell out. Appropriate correction is required.

Claim Rejections - 35 USC § 112

3. Claims 5 and 18 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contain subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The method of "***storing the shadow copy in the low latency cache memory with the microprocessor if the task priority register is accessed frequently***" is not described in the specification to enable one having ordinary skill in the art to understand the invention as claimed. Applicant is advised to amend the specification or cancel the limitation from the claim. Applicant is also reminded that no new matter should be added in the amendment.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1 – 4, 6, 8 – 11, 14 – 17, 19, 21 – 23 and 26 - 29 are rejected under 35 U.S.C. 103(a) being unpatentable over Williams et al (Williams) in view of "Intel Itanium Processor Family Interrupt Architecture Guide"(Intel).

As per claim 1, Williams discloses:

- receiving a command to write an interrupt mask value to the task priority register; (Col 4 lines 20 – 21, "The interrupt logic is configured for updating the interrupt status value in the interrupt register") having the interrupt configured means being able to receive commands to write(update) interrupt value.

However Williams does not disclose, ***writing the interrupt mask value to the task priority register.***

On the other hand, Intel discloses:

- writing the interrupt mask value to the task priority register; (Page 2-4 lines 13 – 14, "Software writes to this register when it is ready to accept another interrupt at a higher or same priority level than specified by the TPR").

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the teaching of Intel into the method of Williams to write the interrupt mask value in the task priority register. The modification would have been obvious

because one of the ordinary skills of the art would have found it motivated to write the interrupt mask value to the task priority register to specify the priority level of the interrupt.

As per claim 1, Williams further discloses:

- writing the interrupt mask value into a shadow copy of the task priority register, wherein the shadow copy is written each time that the task priority register is written. (col 3 lines 12 – 14, "copying the interrupt status value to a prescribed location in the system memory based on detecting at least one interrupt condition").

As per claim 2, the rejection of claim 1 is incorporated and further Williams discloses:

- receiving a command to read the interrupt mask value from the task priority register (col 2 lines 48 – 49, "the I/O device generates an interrupt to notify the CPU of an interrupt condition requiring servicing").

- reading the interrupt mask value from the shadow copy, instead of from the task priority register. (col 4 lines 34- 36, "enabling the CPU 12 to read the interrupt status value from the system memory location 30a in response to an interrupt").

As per claim 3, the rejection of claim 1 is incorporated and further Williams discloses:

- wherein the shadow copy is always written after the task priority register is written. (col 5 lines 6 – 11, "the interrupt logic 32 updates in step 56 the interrupt status value by setting to "1" the selected bit that specifies the corresponding interrupt condition. The interrupt logic 32

then copies the interrupt status value stored in the interrupt register 34 into the assigned system memory location").

As per claim 4, the rejection of claim 3 is incorporated and further Williams discloses:

- further comprising, upon receiving an interrupt, reading the interrupt mask value from the task priority register and writing the interrupt mask value to the shadow copy (col 4 lines 31- 34, "upon updating the interrupt status value in the interrupt register 34, copies the interrupt status values to the system memory location 30a specified by the system memory address register").

As per claim 6, the rejection of claim 1 is incorporated and further Williams discloses:

- wherein the method obviates a need to use a serialize instruction after the task priority register is written because each interrupt performs a serialize operation and performs a read of an interrupt vector register (col 4 lines 31 – 34, "upon updating the interrupt status value in the interrupt register 34, copies the interrupt status values to the system memory location 30a specified by the system memory address register") updating the interrupt status value means updating the interrupt priority before copying to the system memory register and that avoids any further serialization as it is updated before copying.

As per claim 8, the rejection of claim 2 is incorporated and further Williams discloses:

- whereby a latency of reading from the task priority register is substantially reduced. (col 4 lines 49 – 51, "minimizing 110 read operations by the CPU 12, based on copying interrupt status values into the system memory").

As per claim 9, the rejection of claim 1 is incorporated and further Intel discloses:

- ***wherein the microprocessor comprises an IPF architecture microprocessor*** (page 1-1 lines 3 – 5, "Intel® Itanium® architecture Streamlined Advanced Programmable Interrupt Controller (SAPIC). SAPIC is the high performance interrupt architecture for the Itanium architecture"), Intel Itanium architecture is Itanium Processor Family architecture (IPF).

As per claim 10, the rejection of claim 9 is incorporated and further Intel discloses:

- ***wherein the local PIC unit in the microprocessor comprises a streamlined advanced programmable interrupt controller (SAPIC) unit in the IPF architecture microprocessor*** (page 2-1 lines 5 – 7, "The portion that resides in the processor is known as the Local SAPIC unit, or Local SAPIC, and its primary responsibility is to receive, accept, and process the interrupts" and lines 8-10, "The portion that resides in the I/O subsystem is known as the I/O xAPIC unit, or I/O xAPIC, and its only responsibility is to generate interrupts directed to each selected processor's Local SAPIC unit" and fig 2-1.), local APIC unit is local advanced programmable interrupt control unit.

As per claim 11, the rejection of claim 1 is incorporated and further incloses:

- ***wherein the method is performed by an operating system*** (col 2 lines 59 – 61, " a computing system having a central processing unit (CPU), a system memory, and an Input/Output (I/O) device configured for accessing the system memory via a peripheral bus") a computing system inherently includes an operating system.

As per claim 26, Williams discloses:

- receiving a command to read an interrupt mask value from the task priority register (col 2 lines 48 – 49, "the I/O device generates an interrupt to notify the CPU of an interrupt condition requiring servicing").

- reading the interrupt mask value from the shadow copy at a memory location, instead of from the task priority register itself. (col 4 lines 34- 36, "enabling the CPU 12 to read the interrupt status value from the system memory location 30a in response to an interrupt").

As per claim 27, Williams discloses:

- an operating system with reduced latency to read a task priority register of a local programmable interrupt controller unit within a microprocessor, the operating system comprising microprocessor-executable code configured read the interrupt mask value from the shadow copy at a memory location, instead of from the task priority register itself. (col 3 lines 51 – 56, "computing system 10 includes a central processing unit (CPU) 12, a system controller 14, a system memory 16 such as an SDRAM, and an I/O device 18 configured for accessing the system memory 16 via a peripheral bus 20," And col 4 lines 34- 36, "enabling the CPU 12 to read the interrupt status value from the system memory location 30a in response to an interrupt") a computer system inherently includes an operating system.

As per claim 28, Williams discloses:

- a memory system, including local cache memory on each microprocessor and a main memory, wherein the memory system holds data including an operating system and shadow

copies of the TPRs, and wherein the operating system includes executable-code for reading the interrupt mask values from the shadow copies and for maintaining the shadow copies. (col 3 lines 51 – 56, "computing system 10 includes a central processing unit (CPU) 12, a system controller 14, a system memory 16 such as an SDRAM, and an I/O device 18 configured for accessing the system memory 16 via a peripheral bus 20," And col 4 lines 34- 36, "enabling the CPU 12 to read the interrupt status value from the system memory location 30a in response to an interrupt")

However Williams does not disclose, ***a plurality of microprocessors interconnected by a processor bus.***

On the other hand Intel discloses:

- a plurality of microprocessors interconnected by a processor bus, wherein each microprocessor includes a task priority register (TPR) with a interrupt mask value for that microprocessor (Page 2-1 fig 2-1 and page 2-10 lines 15-16, "Within the processor Local SAPIC unit, there is a Task Priority Register (TPR), which is used by software to mask interrupts based on "priority class."")

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the teaching of Intel into the method of Williams to have a plurality of microprocessor interconnected with a processor bus. The modification would have been obvious because one of the ordinary skills of the art would have found it motivated to have a plurality of microprocessor interconnected with a processor bus for faster access and reduce interrupt execution process and utilize multi-processor resources.

As per claim 29, Williams discloses:

- upon receiving a command to write an interrupt mask value to the task priority register, writing the interrupt mask value to the task priority register without performing a serialization directly thereafter (Col 4 lines 20 – 21, "The interrupt logic is configured for updating the interrupt status value in the interrupt register")

However Williams does not disclose, ***writing the interrupt mask value to the task priority register.***

On the other hand Intel discloses:

- writing the interrupt mask value to the task priority register; (Page 2-4 lines 13 – 14, "Software writes to this register when it is ready to accept another interrupt at a higher or same priority level than specified by the TPR")

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the teaching of Intel into the method of Williams to write the interrupt mask value in the task priority register. The modification would have been obvious because one of the ordinary skills of the art would have found it motivated to write the interrupt mask value to the task priority register to specify the priority level of the interrupt.

As per claim 29, Williams further discloses:

- upon receiving an interrupt, performing the serialization and reading an interrupt vector register, wherein a spurious indicator is returned if the interrupt is maskable (col 5 lines 7 – 10, "the interrupt logic 32 updates in step 56 the interrupt status value by setting to "1" the selected bit that specifies the corresponding interrupt condition. The interrupt logic 32 then

copies the interrupt status value stored in the interrupt register" and col 5 lines 15 – 20, "The CPU 12 in step 62 detects the interrupt, disables interrupts by the peripheral device 18, and reads the interrupt status value from the assigned location 30a in the system memory 16. The CPU 12 performs in step 64 the necessary servicing of the interrupt based on reading the interrupt status value from the location") updating the interrupt status value means updating the interrupt priority (serialization).

Claims 14 – 17, 19 and 21 - 23 are product claims with operating system of claims 1 – 4, 6 and 8 – 11 above. They are therefore rejected under the same rational.

6. Claims 5, 7, 18 and 20 as best understood by the examiner are rejected under 35 U.S.C. 103(a) being unpatentable over Williams et al (Williams) in view of "Intel Itanium Processor Family Interrupt Architecture Guide"(Intel) and further in view of Demharter (Demharter) US Patent 7080205.

As per claim 5, Williams in view of Intel discloses all the elements of claim 5 except ***stored in low-latency cache memory within the microprocessor.***

On the other hand Demharter discloses:

- if the task priority register is accessed frequently, then the shadow copy is stored in low-latency cache memory within the microprocessor. (col 1 lines 66 – 67, "separate cache memories, of which one, for example, is suitable for storing the interrupt vectors and interrupt handlers")

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the teaching of Demharter into the combined method of Williams and Intel to store interrupt value in cache memory. The modification would have been obvious because one of the ordinary skills of the art would use cache as interrupt storage for faster access and execution.

As per claim 7, the rejection of claim 6 is incorporated and further Demharter discloses:

- whereby a latency of writing to the task priority register is substantially reduced. (col 2 lines 8 – 10, "Storage of the interrupt vectors and interrupt handlers in a cache ensures significant reduction in the processing time of an interrupt")

Claims 18 and 20 are product claims with operating system of claims 5 and 7 above. They are therefore rejected under the same rational.

7. Claims 12, 13, 24 and 25 are rejected unders 35 U.S.C. 103(a) being unpatentable over Williams in view of "Intel Itanium Processor Family Interrupt Architecture Guide"(Intel) and further in view of "The Future of the OS for Internet Applications" by Reed Hellman(Hellman).

As per claim 12, Williams in view of Intel discloses all the elements of claim 12 except ***wherein the method is performed by a UNIX-type operating system configured for the IPF architecture microprocessor.***

On the other hand Hellman discloses:

- wherein the method is performed by a UNIX-type operating system configured for the IPF architecture microprocessor (page 3 col 3 lines 2 – 9 , "Windows 2000 continues the Microsoft OS's ability to run on any vendor's Intel-compatible machine. Many vendors' Unix flavors, on the other hand, run only on the company's own hardware, although Solaris 8 will run on Intel's upcoming Itanium chips, as well as Sparc-based systems.").

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the teaching of Hellman into the combined method of Williams and Intel to use UNIX operating system with IPF architecture microprocessor. The modification would have been obvious because one of the ordinary skills of the art would have found it motivated to use UNIX based operating system because it has evolved in a powerful operating system and also it is less machine specific than many other operating systems.

As per claim 13, Williams in view of Intel discloses all the elements of claim 13 except ***wherein the method is performed by a Windows operating system configured for the IPF architecture microprocessor.***

On the other hand Hellman discloses:

- wherein the method is performed by a version of a Windows operating system configured for the IPF architecture microprocessor. (page 3 col 3 lines 2 – 9 , "Windows 2000 continues the Microsoft OS's ability to run on any vendor's Intel-compatible machine. Many vendors' Unix flavors, on the other hand, run only on the company's own hardware, although Solaris 8 will run on Intel's upcoming Itanium chips, as well as Sparc-based systems.").

Therefore, it would have been obvious to a person of ordinary skill in art at the time of invention was made to incorporate the Hellman into the combined method of Williams and Intel to use Windows operating system with IPF architecture microprocessor. The modification would have been obvious because one of the ordinary skills of the art would have found it motivated to use Windows based operating system because it is a user friendly multitasking graphical user interface self contained operating system with drop down menu, windowed region screen and pointing device very widely used for desktop computers.

Claims 24 and 25 are product claims with operating system of claims 12 and 13 above. They are therefore rejected under the same rational.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

TITLE: Input/output device configured for minimizing I/O read operations by copying values to system memory, US 6,665,750 B1

TITLE: Arrangement and method for reducing the processing time of a data processing device, US 7,080,205 B2

TITLE: Multiprocessor computer system with data bus and ordered and out-of-order split data transactions, US 5,191,649

TITLE: Real time interrupt handling for superscalar processors, US 6,295,574 B1

TITLE: Interrupt distribution scheme for a computer bus, US 5,282,272

TITLE: Intel® Itanium® Processor Family Interrupt Architecture Guide, March 2003

TITLE: The Future of the OS for Internet Applications, Reed Hellman, IEEE, Volume 33 Issue 5 May 2000

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Abdullah-Al Kawsar whose telephone number is 571-270-3169. The examiner can normally be reached on 7:30am to 5:00pm, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chameli Das can be reached on 571-270-1392. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Abdullah-Al Kawsar

JEAN M. CORRIELUS
PRIMARY EXAMINER
Art unit 2162
5/24/07